

Banco de Dados Relacionais

▶ Sentenças SQL

// Conceitos de linguagem de consulta de banco de dados

O que é SQL?

SQL (Structured Query Language): é uma linguagem padrão usada para criar, modificar, manter e consultar um banco de dados relacional.

É um padrão ANSI, isso implica que conhecendo essa linguagem você pode manipular qualquer banco de dados de qualquer fabricante (claramente, cada fabricante tem suas particularidades e extensões, mas a sintaxe é semelhante em todos os casos).

DDL e DML

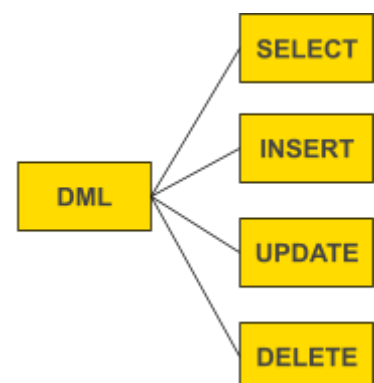
SQL pode ser dividido em duas partes DDL e DML. Cada parte possui instruções/comandos para realizar operações de natureza diferente. Esses dois grupos são:

DDL (Data Definition Language): Eles nos permitirão criar novos bancos de dados, modificá-los e excluí-los. Esses comandos afetam tabelas, campos e índices.

- CREATE - Cria novas tabelas, campos e índices.
- ALTER - Modifica as tabelas
- DROP - Elimina tabelas e índices.

DML (Data Manipulation Language): Nos permite manipular, consultar, classificar, filtrar, etc. dados existentes em um banco de dados.

- SELECT - Consulta registros no banco de dados.
- INSERT - Insere um novo registro
- UPDATE - Modifica valores de um registro
- DELETE - Elimina registros



SELECT

Esta instrução é usada para recuperar/exibir dados de uma tabela.

O formato geral ao usar este comando é:

```
SELECT <colunas separadas por vírgulas> FROM <tabela>;
```

Tendo em conta que existe uma tabela de Clientes:

```
Cliente (id_cliente, rg, nome, sobrenome, endereco);
```

Exemplos:

```
SELECT * FROM Cliente;
```

//Recupera todas as colunas da tabela; sem definir a cláusula WHERE, todos os registros da tabela Cliente são obtidos.

```
SELECT nome FROM Cliente;
```

//Recupera todos os nomes da tabela Cliente.

```
SELECT sobrenome, endereco FROM Cliente;
```

//Recupera todos os sobrenomes e os endereços da tabela.

WHERE: Esta declaração é usada para propor a condição específica que os dados a serem recuperados devem atender.

Exemplo:

```
SELECT sobrenome, endereco FROM Cliente WHERE nome='Paolo';
```

//Desta forma, estaria mostrando o sobrenome e endereço de todos os clientes com o nome Paolo.

INSERT

Esta instrução é usada para adicionar dados a uma tabela.

O formato geral ao usar este comando é:

```
INSERT INTO <tabela> VALUES (dado 1, dado 2, dado n);
```

Exemplos:

Considerando que a tabela Clientes tem a seguinte estrutura:

```
Cliente (id_cliente INTEGER, rg VARCHAR, nome VARCHAR, sobrenome VARCHAR, endereco  
VARCHAR);
```

Para inserir linhas na tabela:

```
INSERT INTO Cliente VALUES (0001, '00111000', 'Andriy', 'Shevchenko', 'Ucrania 1613');
```



INSERT INTO Cliente **VALUES** (0002, '00999111', 'Andrea', 'Pirlo', 'Italia 2006');
INSERT INTO Cliente **VALUES** (0003, '00111222', 'Paolo', 'Maldini', 'Milan 2007');

UPDATE

Esta instrução é usada para modificar dados em uma tabela.

O formato geral ao usar este comando é:

UPDATE <tabela> **SET** dadoaModificar='NovoValor' **WHERE** dado='Valor';

Exemplo:

UPDATE Cliente **SET** endereco='Turín 2012' **WHERE** rg='00999111';



Se não colocarmos a cláusula **WHERE**, todos os endereços da tabela Cliente serão modificados.

DELETE

Esta instrução é usada para excluir registros em uma tabela.

O formato geral ao usar este comando é:

DELETE FROM <tabela> **WHERE** dado='Valor';

Exemplo:

DELETE FROM Cliente **WHERE** id_cliente=0001;



Se não colocarmos a cláusula **WHERE**, todos os registros da tabela Clientes serão deletados.